

АНАЛИЗ И МЕТОДИКА РЕШЕНИЯ ПРОБЛЕМЫ КРИТИЧЕСКОЙ НАГРУЗКИ БАЗ ДАННЫХ MySQL

Бондаренко В.Ю., Борис Т.В.

Научный руководитель: канд. техн. наук Алексеев Н.А.

Национальный технический университет Украины «КПИ», Украина

E-mail: volodymyr.bondarenko.ua@gmail.com

Abstract — Работа посвящена анализу наиболее распространенных причин критической нагрузки в базах данных MySQL и способов их предотвращения.

1. Введение

Успешная работа веб приложений напрямую зависит от конфигурации системы управления базами данных. В настоящее время одной из самых распространенных СУБД для создания веб-сайтов является бесплатный программный продукт OracleMySQL. Однако, в процессе функционирования таких сайтов возникают проблемы потери производительности СУБД, особенно с ростом числа запросов, работающих с высокими объемами данных. Данная работа посвящена анализу наиболее часто встречающихся причин возникновения критической нагрузки базы MySQL и способов их устранения.

2. Основная часть

В ходе работы над созданием веб приложения LookForFilm.net, в котором обрабатываются большие массивы данных, были накоплены обширные статистические данные случаев возникновения критической нагрузки. В базе данных под управлением MySQL сервис хранит архивы фильмов, портфолио актеров и их взаимосвязи, оценки пользователей, а также формирует индивидуальные рейтинги фильмов для будущего просмотра.

В ходе анализа ситуаций увеличения нагрузки на сервер были отмечены различные проблемы с системой управления базами данных. В то время как другими вычислительными процессами виртуального сервера потреблялось не более 5 % WCPU, процесс *mysqld* полностью занимал доступное процессорное время. Такая критическая нагрузка способствовала отказу сервера, что в свою очередь делало веб приложение недоступным для работы. Такая ситуация возникала по ряду причин, анализ которых позволил успешно реализовать алгоритмы взаимодействия приложения с базой данных, позволившие избежать неработоспособности в случае возникновения критической нагрузки СУБД MySQL:

— большое количество запросов к базе данных. Опираясь на результаты проведенных в данной работе исследования проблемы возникновения критической нагрузки на СУБД веб-приложений вследствие роста количества запросов к базе данных рекомендуется решать внедрением технологии кэширования повторяющихся результатов запросов СУБД, тем самым снижая соответствующую нагрузку. Рассмотрев основные сильные и слабые стороны существующих решений, было принято решение использовать *Memcached*, который обеспечивает универсальный подход к кэшированию информации;

— большое количество кортежей в таблицах. Одним из способов решения этой проблемы является реализация набора индексов, т.е. создание схем таблицы с использованием только тех атрибутов, которые часто используются в *SELECT* запросах. Данный подход является эффективным, только при условии преобладания *SELECT* запросов (наличие индексов замедляет редактирование данных) и огра-

ниченного количества кортежей. Более эффективным является логическое разделение данных по подтаблицам используя встроенные функции партиционирования MySQL. Проведенные тесты, однако, показали, что прирост производительности при использовании встроенных возможностей MySQL меньше по сравнению с формированием подтаблиц непосредственно в процессе сохранения данных самим веб-приложением. Использование такого подхода позволяет добиться положительной динамики работы сервиса при масштабировании данных;

— большое количество одновременных запросов к базе данных. Для подсчета индивидуального рейтинга фильмов тестируемый ресурс должен был обрабатывать от 10 тыс. до 100 тыс. SQL запросов. Такое количество моментальных запросов способно полностью загрузить базу данных. Был протестирован и внедрен универсальный способ разбивания обработки данных на *n* итераций. Практически это было реализовано на языке *JavaScript*;

— использование группировки и сортировки кортежей в таблицах. После детального анализа стало понятно, что некоторые группировки и сортировки быстрее будут выполняться на стороне веб приложения [1]. Таким образом, для достижения максимального быстродействия необходимо каждый запрос проверять на целесообразность применения сортировок и группирования элементов непосредственно в базе.

3. Заключение

Анализ фактов возникновения критической нагрузки на базы данных веб-приложений, использующих СУБД MySQL позволил выделить наиболее часто встречающихся их причины. Рассмотренные в работе технические решения могут служить универсальной методикой их устранения.

4. Список литературы

- [1] Aliksieiev M. Analysis of approaches used to raise productivity of PHP programs / M. Aliksieiev, V. Bondarenko // Proc. of conf «Modern Problems of Radio Engineering Telecommunications and Computer Science» (TCSET'2012). — Lviv-Slavske: Publishing House of Lviv Polytechnic, 2012. — P. 349.

ANALYSIS AND METHODOLOGY OF SOLVING OF THE PROBLEM OF A CRITICAL LOADING IN DATABASE MySQL

Bondarenko V.U., Boris T.V.

Scientific adviser: Alekseev N.A.

National Technical University of Ukraine
"Kyiv Polytechnic Institute", Ukraine

Abstract — The analysis of the most common causes of a critical load in MySQL databases and ways of their prevention or elimination is presented.